INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Toward a geometric view on computations

Philippe Audebaud

N° 5492

February 2005

Thème SYM

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--5492--FR+ENG

Toward a geometric view on computations

Philippe Audebaud*

Thème SYM — Systèmes symboliques
Projet Lemme

Rapport de recherche n° 5492 — February 2005 — 16 pages

Abstract: We interpret Intersection Types as the closed sets of some Zariski topology on pure λ-terms. In this view, the *parallel or* operator introduced by Boudol is the multiplication for an underlying ring structure. We propose a new calculus which extends pure λ-calculus along the same lines as relative numbers $\mathbb{Z}$ extend natural numbers $\mathbb{N}$, the ring operations expressing computation rules on terms.

Thus, types are interpreted as the zeros sets for some notion of polynomial ideals (algebraic sets). Terms properties (strong normalisation, confluence, full abstraction) are investigated. Among similarities with Algebraic Geometry, we suggest that terms of interest, such as normalising terms or convergent programs are rare; divergence is a generic property for programs.

Key-words: algebraic geometry, lambda calculus, full abstraction, model

* ENS Lyon and INRIA Sophia-Antipolis

Vers une lecture géométrique des calculs

Résumé : Nous interprétons les types intersection comme des fermés pour une topologie de Zariski sur les λ-termes. Dans cette lecture, le *ou parallèle* de Boudol est la multiplication d'une structure d'anneau sous-jacente. Nous proposons un nouveau calcul qui étend le λ-calcul pur et se compare à ce dernier comme l'anneau des entiers relatifs se comparent aux entiers naturels.

Ainsi, les types sont vus comme l'ensemble des zéros d'idéaux de polynômes (ensembles algébriques). Nous étudions certaines propriétés connues comme la confluence, la normalisation forte et le problème de l'adéquation complète dans ce nouveau contexte. Parmi les similitudes avec la géométrie algébrique que nous pouvons suggérer, la normalisation forte ou la convergence des programmes sont des pripriétés rares ; leur opposées doivent être considérées comme des propriétés génériques des termes étendus.

Mots-clés : géométrie algébrique, lambda calcul, full abstraction, modèle

Toward a geometric view on computations

Philippe Audebaud

February 4, 2005

e interpret Intersection Types as the closed sets of some Zariski topology on pure λ-terms. In this view, the *parallel or* operator introduced by Boudol is the multiplication for an underlying ring structure. We propose a new calculus which extends pure λ-calculus along the same lines as relative numbers $\mathbb{Z}$ extend natural numbers $\mathbb{N}$, the ring operations expressing computation rules on terms.

Then, types are interpreted as the zeros sets for some notion of polynomial ideals (algebraic sets). Terms properties (strong normalisation, confluence, full abstraction) are investigated. Among similarities with Algebraic Geometry, we suggest that terms of interest, such as normalising terms or convergent programs are rare; divergence is a generic property for programs.

Introduction

Type assignment systems (t.a.s.) have proved powerful when studying properties of pure λ-terms [1, 2, 8, 7]. In this view, types are meant as formulae, logical properties of terms. If logical formulae are assigned to terms on the syntactic side, which meaning do have this relationship, from the semantic side? Let us restrict the query to concrete interpretations as subsets of some topological space. It is usely accepted that formulae are interpreted as open sets. Stone [17] and Tarski [18] discovered that topological spaces allow for complete models of intuitionistic propositional logic. Later, Smyth [16] (see also Abramsky [1]) remark that open sets are analogous to semi-decidable properties advocated for this approach. Holmes [13] although emphasises a the converse view as closed sets. Let us have a closer look in the specific area of t.a.s.. Assume types σ are generated by the following grammar

$$\sigma \quad ::= \quad \omega|t|\sigma \wedge \sigma|\sigma \to \sigma$$

where t ranges over a denumerable set of type variables, and come together with some preorder relation $\leqslant$. We recall that a topology $\mathcal{O}$ on Λ is a family of subsets of Λ containing the empty set and Λ, closed under finite intersections and arbitrary unions; the family $\mathcal{F} \equiv (\Lambda \setminus o)_{o \in \mathcal{O}}$ represent the closed sets for this topology, and is in turn closed under arbitrary intersections and finite unions. How precise the description of the topology should be ?

Given any interpretation $[t] \subseteq \Lambda$ for each type variable t, it is extended inductively by

$$[\sigma \wedge \tau] = [\sigma] \cap [\tau]$$

and

$$[\sigma \to \tau] \equiv [\sigma] \to [\tau] \equiv \{e|\forall a \in [\sigma].(ea) \in [\tau]\} = \cap_{a \in [\sigma]}\{e|(ea) \in [\tau]\}$$

*ENS Lyon and INRIA Sophia-Antipolis

A preliminary observation is that $F_a : x \mapsto (xa)$ is the sole construction involved on terms. This is a key point in every normalisation proof by way of reducibility candidates. Next, for the interpretation to be closed by finite or arbitrary intersections it is sufficient to require that $F_a^{-1}(V))$ of any closed subset V is still closed. Therefore we do expect *at least* these applications to be continuous with respect to the topology we are looking forward.

Take $[t] = \pi \in \mathcal{F}$ for all t, which is a fair assumption since we cannot discriminate within type variables. Then, for all σ, $[\sigma] \in \mathcal{F}$ is built as an arbitrary intersection of $F_{\bar{u}}^{-1}(\pi)$ where $\bar{u} \in \bar{\Lambda}$ range over sequences of terms. Let us take a fresh symbol X as variable, and define $(X\bar{u})$ to be a *monomial*. Let us say $e \in \Lambda$ *cancels* the monomial $M(X) = (X\bar{u})$ if $M(e) = (e\bar{u}) \in \pi$. Now, $[\sigma] = V(I) \equiv \{e | \forall M \in I. M(e) \in \pi\}$ is expressed as the set of (common) zeroes of some family I of monomials. Given a family $(I_k)_k$ of sets of monomials, $\cap_k V(I_k) = V(\cup_k I_k)$, so the collection of such sets of closed under arbitrary intersection. For this collection to represent the closed sets of some topology on Λ, it requires also $V(I) \cup V(J)$ to be identified with some $V(K)$, where K is some set of monomials depending on I and J. By definition, we must have $K \subseteq I \cap J$. But, since $e \in V(I) \cup V(J)$ iff $e \in V(I)$ or $e \in V(J)$, there is no way to find such a K if we stay on our former collection of monomials. A possible approach would be to assert a priori the collection of $V(\{M\})$ to form the *sub-basis* of elementary closed subsets for the topology we are looking ahead (see [19] for example). Anyway, this does not provide a concrete construction for K.

Let us instead extend the monomial definition with a new operation $M_1 * M_2$ and a new symbol 1 in such a way our monomials now form a (multiplicative commutative monoid for the$*$law. We take K to be the set of all monomials $M_I * M_J$, where $M_I \in I$ and $M_J \in J$, shortened $K = I * J$. Then $V(I) \cup V(J) \subseteq V(I \cap J)$. Since $I * J \subseteq I \cap J$, we get $V(I \cap J) \subseteq V(I * J)$. If $e \notin V(I) \cup V(J)$, there exists M_I and M_J such that $M_I(e) \notin \pi$ and $M_J(e) \notin \pi$. Introduce the operation$*$on terms as well together with the constant 1 and let $\mathbb{E} \supseteq \Lambda$ for the resulting set. Let us assume π satisfies the property:

$$(\pi\,\text{prime}) \qquad \forall a, b. a * b \in \pi \text{ implies } a \in \pi \text{ or } b \in \pi$$

Under this assumption, we get $M_I(e) * M_J(e) \notin \pi$, hence the condition $e \notin V(I * J)$ which ensures $V(I) \cup V(J) = V(I * J)$. For the empty set to be represented this way, it is sufficient to assume $1 \notin \pi$ since $\varnothing = V(\{1\})$. Last, $\mathbb{E} = V(\{e\})$ as soon as there exists some $e \in \pi$, taken into account as some constant monomial.

To sum up, provided the extended calculus $\mathbb{E}$ and the various conditions on $\pi \subseteq \mathbb{E}$, types can be interpreted as closed sets of some particular topology for which any closed set is the set $V(I)$ of common zeroes for a family of monomials. The reader would have recognised that we come close to the definition of algebraic sets as zeroes sets of polynomial ideals in the context of Algebraic Geometry. Before going any further in this direction, let us ask ourselves whether it is worthwhile? Assume π denote the set of convergent terms for some convergence predicate $\bullet \Downarrow$. The (π prime) condition is translated into, whenever $a * b \Downarrow$, then $a \Downarrow$ or $b \Downarrow$. In this particular reading, the $a * b$ coincides with the parallel construct $a \| b$ introduced by Boudol [4] . However, it is worth noticing that our$*$operation does not rely on any intended meaning, as far as syntax is concerned. More precisely, our understanding is that the parallel behaviour is rather a *property* than part of the definition of this operator. Instead, since the λ-calculus carries precisely the notion of computation, the new operation $a * b$ should represent a new computation scheme over terms, called thereafter *multiplication*.

Yet, the set $\mathbb{E}$ is not a ring, the *addition* operation being missing. So let us introduce $a + b$ together with the new constant 0 such that the resulting *extended calculus* $\mathbb{E}$ turns into a commutative unitary ring, embedding Λ. (Informally, $\mathbb{E}$ compares itself to the ring of relative numbers $\mathbb{Z}$ generated by natural numbers $\mathbb{N}$, λ-abstraction being the major source of expressiveness... and problems.) The particular case when π collects convergent terms leads naturally to the condition $(a+b) \Downarrow$ if $a \Downarrow$ and $b \Downarrow$. In that case, addition is interpreted as a conjunction as in [10]. (Beware that this definition is sensible: if $a \Uparrow$, then $a + (-a)) \Uparrow$, while it can be simplified to 0! Therefore, we cannot argue up to algebraic equalities.) More abstractly, the whole conditions on π express the fact that it has to be a prime and proper ideal of the ring $\mathbb{E}$. When $\pi = \langle 0 \rangle$ is the ideal generated by 0, the corresponding topology is known as the Zariski topology, where polynomials are simply finite sums of the previously described monomials. This observation emphasises the additional condition we put on $\mathbb{E}$ to be an *integral domain*: $0 \neq 1$ and the ideal $\langle 0 \rangle$ is prime.

Our goal in this article is simply to collect the basics properties of the extended calculus $\mathbb{E}$ and to ensure that we are provided with a model rich enough to study term properties, which was our initial motivation. We shall concentrate ourselves on strong normalisation, confluence and full abstraction problems.

Section 1 presents the calculus, the reduction relation and typing judgement. Polynomials and the few elements from algebra required to interpret types are object of section 2. Next, we study strong normalisation and confluence along the lines of our algebraic model. The section 4 addresses the full abstraction problem. Discussion of further matters concludes this article.

1 The computations ring $\mathbb{E}$

To save space, we lean on introduction for the presentation of the material and address the definitions with no further explanation. The extended terms (terms for short) are defined by the following grammar

$$e \quad ::= \quad x|\lambda x.e|(ee)|0|1|e+e|-e|e*e,$$

where x ranges a given denumerable set of variables Var.

We keep the current usage, both from the pure λ-calculus where application associates to the left, and from the mathematical structures, where operations $+$ and * associate to the left and * has stronger precedence than $+$. Moreover, λ-calculus operations (abstraction and application) have precedence over ring operations ($+$, *). As usual, $a - b$ stands for $a + (-b)$.

In this article, both structure cohabit, as different layers over terms. For example in $(\lambda x.(x + x)\lambda y.y)$ the out-most layer is an application, while $1 + \lambda x.x$ is a ring expression. As a matter of consequence, most definitions from Commutative Algebra will keep identical hereafter.

If $A \subseteq \mathbb{E}$ is any set of terms, A is an *ideal* if it is both a (commutative) additive subgroup of E and contains every product $a * e$ where $a \in A$ and $e \in E$. A is a sub-ring if it is both an additive subgroup of $\mathbb{E}$ and an unitary monoid with respect to product. Given $S \subseteq \mathbb{E}$, the ideal generated by S, denoted $\langle S \rangle$, is the smallest ideal which contains S; it is equally defined as the intersection of all the ideals containing S, since this construction preserves the ideal structure.

1.1 Reductions

Besides the usual β-reduction, we introduce two sets of reduction rules. *Algebraic reductions* consist in left to right orientation of the usual equations, expressing $(\mathbb{E}, +, *)$ being a commutative unitary ring:

$$
\begin{array}{rl}
(0+) & e + 0 \rightarrow e \\
(C+) & e_1 + e_2 \rightarrow e_2 + e_1 \\
(A+) & (e_1 + e_2) + e_3 \rightarrow e_1 + (e_2 + e_3) \\
(I+) & e_1 + (-e_1) \rightarrow 0 \\
(-0) & -0 \rightarrow 0 \\
(--) & -(-e) \rightarrow e \\
(-+) & -(e_1 + e_2) \rightarrow (-e_1) + (-e_2) \\
(0*) & e * 0 \rightarrow 0 \\
(1*) & e * 1 \rightarrow e \\
(C*) & e_1 * e_2 \rightarrow e_2 * e_1 \\
(A*) & e_1 * (e_2 * e_3) \rightarrow e_1 * (e_2 * e_3) \\
(D) & e * (e_1 + e_2) \rightarrow e * e_1 + e * e_2 \\
(*-) & e_1 * (-e_2) \rightarrow -(e_1 * e_2)
\end{array}
$$

The set of algebraic reductions is denoted ALG. The notation $e \rightarrow_{\text{alg}} e'$ will be used for the reflexive and transitive closure of this set of reductions.

The second set of reduction rules, denoted HOM captures the homomorphic nature of $\bullet \mapsto (\bullet c)$ and the fact that 0 and 1 are overloaded as constant functions in the context of λ-calculus:

$$
\begin{array}{rl}
(*\,\text{app}) & (a * bc) \rightarrow (ac) * (bc) \\
(+\,\text{app}) & (a + bc) \rightarrow (ac) + (bc) \\
 & \\
(-\,\text{app}) & (-ac) \rightarrow -(ac) \\
(1\,\text{app}) & (1c) \rightarrow 1 \\
(0\,\text{app}) & (0c) \rightarrow 0
\end{array}
$$

So far for the reduction rules. We let $\twoheadrightarrow$stands for binary relation which is the transitive closure of the pre-congruence built on all the above reduction relations.

Proposition 1 (Confluence) *The relation $\twoheadrightarrow$enjoys confluence: for all $e, e_1, e_2 \in \mathbb{E}$, whenever $e \twoheadrightarrow e_1$ and $e \twoheadrightarrow e_2$ there exists e_3 such that $e_1 \twoheadrightarrow e_3$ and $e_2 \twoheadrightarrow e_3$.*

Proof Variant of the usual Tait-Martin Löf method, along the explanations given in [10]. □

This property will be discussed also at the end of section 4.

1.2 The set of types and its preorder.

Given a denumerable set $\mathsf{TypeVar}$ of type variables, ranged over by t, and the type constant ω, we define the set Type of types σ generated by the following grammar

$$\sigma \quad ::= \quad \omega | t | \sigma \wedge \sigma | \sigma \rightarrow \sigma$$

The set Type is provided with the preorder $\leqslant$ presented as the smallest binary relation such that

i. $\leqslant$ is both reflexive and transitive;

ii. $\sigma \wedge \tau \leqslant \rho$ iff $\sigma \leqslant \rho$ and $\tau \leqslant \rho$;

iii. for all type σ, $\sigma \leqslant \omega$;

iv. $\sigma' \leqslant \sigma$ and $\tau \leqslant \tau'$ imply $\sigma \to \tau \leqslant \sigma' \to \tau'$;

v. $\sigma \to \omega \leqslant \omega \to \omega$;

vi. $(\sigma \to \tau) \wedge (\sigma \to \rho) \leqslant \sigma \to (\tau \wedge \rho)$.

The rule (v) is worth noticing. It is not a particular instance of rule iii). It has been introduced in [4] for the strict calculus. We take this relation into account in this article for expressing the idea that $\omega \to \omega$ retains the ***least piece of information*** that we could get for a term; typed within ω otherwise. In other words, let $=$ stands for the symmetric closure of $\leqslant$. Then $\sigma \neq \omega$ iff $\sigma \leqslant \omega \to \omega$.

1.3 Type inference

The design of the inference rules is an important matter. In this article, we take the point of view that any type should be interpreted as a set of terms which is invariant under algebraic computations. Thus, a type is meant to be an additive sub-group. As for the multiplication, given a of type σ, and e any term, we expect $e * a$ to be of type σ too. Therefore, a type is meant to be an ideal. This is taken into account by rules $(R \ldots .)$. So the constant 0 belongs to any type. But 1 belongs to none, except ω; assuming otherwise would lead to $e * 1 \in \sigma$ for all e, as soon as $1 \in \sigma$. Hence, $e \in \sigma$, since types are also expected to be stable by reduction.

Notice also that we have restricted the rule (ω) in such a way free variables of term e must be declared in the context Δ for the judgement $\Delta \vdash e : \omega$ to be allowed. The other rules are the usual ones.

$$\text{(Hyp)} \quad \frac{\Delta \vdash (x : \sigma) \in \Delta}{\Delta \vdash x : \sigma}$$

$$(\to I) \quad \frac{\Delta, x : \sigma \vdash e : \tau}{\Delta \vdash \lambda x.e : \sigma \to \tau}$$

$$(\to \mathbb{E}) \quad \frac{\Delta \vdash a : \sigma \to \tau \Delta \vdash b : \sigma}{\Delta \vdash (ab) : \tau}$$

$$(T\omega) \quad \frac{\Delta \vdash}{\Delta \vdash e : \omega} \mathrm{FV}(e) \subseteq \mathrm{FV}(\Delta)$$

$$(T \leqslant) \quad \frac{\Delta \vdash e : \sigma \sigma \leqslant \sigma'}{\Delta \vdash e : \sigma'}$$

$$(R0) \quad \frac{\Delta \vdash}{\Delta \vdash 0 : \sigma}$$

$$(R-) \quad \frac{\Delta \vdash a : \sigma}{\Delta \vdash -a : \sigma}$$

$$(R+) \quad \frac{\Delta \vdash a : \sigma \Delta \vdash b : \sigma}{\Delta \vdash a + b : \sigma}$$

$$(R*) \quad \frac{\Delta \vdash a : \sigma \Delta \vdash b : \tau}{\Delta \vdash a * b : \sigma}$$

$$(T\wedge) \quad \frac{\Delta \vdash e : \sigma \Delta \vdash e : \tau}{\Delta \vdash e : \sigma \wedge \tau}$$

As usual, we write $\Delta \vdash e : \rho$ if there exists a proof ending with this judgement along the lines of the above inference system. Assuming the last rule applied is neither (T$\wedge$), (T$\leqslant$) nor (Tω), exactly one inference rule applies, depending on the structure of term e. Let us define $\Delta \vdash^s e : \rho$ for any derivation which ends with a structural rule.

Proposition 2 (Structural Analysis)

i. *Whenever* $\Delta \vdash e : \rho$, *then* $\exists n, \rho_1 \ldots \rho_n$ *such that* $\rho_1 \wedge \ldots \wedge \rho_n \leqslant \rho$ *and* $\forall i . \Delta \vdash^s e : \rho_i$.

ii. $\Delta \vdash^s 0 : \rho$ *always.*

iii. $\Delta \vdash^s x : \rho$ *iff* $(x : \rho) \in \Delta$.

iv. $\Delta \vdash^s \lambda x . e : \rho$ *iff* $\rho = \sigma \to \tau$ *and* $\Delta, x : \sigma \vdash e : \tau$ *for some* σ, τ.

v. $\Delta \vdash^s (ab) : \rho$ *iff* $\Delta \vdash a : \sigma \to \rho$ *and* $\Delta \vdash b : \sigma$ *for some* σ.

vi. $\Delta \vdash^s a + b : \rho$ *iff* $\Delta \vdash a : \rho$ *and* $\Delta \vdash b : \rho$.

vii. $\Delta \vdash^s -a : \rho$ *iff* $\Delta \vdash a : \rho$.

viii. $\Delta \vdash^s a * b : \rho$ *iff* $\Delta \vdash a : \rho$ *or* $\Delta \vdash b : \rho$.

Proof The sensible part is point i) which actually relies on solving the types inequalities $\gamma \wedge \delta \leqslant \rho \to \tau$ and $(\bigwedge_{i=1}^{n} \gamma_i \to \delta_i) \leqslant \rho \to \tau$. We hide the details which develop themselves as usual in the context of t.a.s.. □

Lemma 1 (Reduction) *If* $\Delta \vdash e : \sigma$ *and* $e \twoheadrightarrow e'$ *then* $\Delta \vdash e' : \sigma$.

2 Zariski Topologies on $\mathbb{E}$

In Algebraic Geometry, Zariski sets are defined as sets of common zeroes for a family of polynomials. In the current context, we have both to define an appropriate notion of polynomial $P(X)$, and provide accordingly a meaning to the relation $P(e) = 0$.

2.1 Polynomials

In the introduction, we saw the importance of applicative terms $(e\bar{u})$. Although very well known, Parigot's strong normalization proof of the $\lambda\mu$-calculus [15] in our understanding of the importance of sets of sequences of termsfor the purpose of building our topology. Later references both based on this idea are [3, 9] among many others.

Let $\mathbb{E}^{<\omega}$ stands for the set of finite sequences $\bar{e}$ of elements of $\mathbb{E}$, extended with a new symbol 0, not to be confused with the empty sequence. Define as $\Sigma_{\mathbb{E}}$ the free commutative monoid generated by $\mathbb{E}^{<\omega}$. For X fresh variable, define $X^0 \equiv 1$, $X^{\bar{e}} \equiv (X\bar{e})$ and $X^{\bar{p}+\bar{q}} \equiv X^{\bar{p}} * X^{\bar{q}}$, for $p, q \in \Sigma_{\mathbb{E}}$. A term $e * X^p$, where $e \in \mathbb{E}$ and $p \in \Sigma_{\mathbb{E}}$, is called a monomial with coefficient $e \in \mathbb{E}$. The degree $d^o(X^p)$ of the monomial X^p is defined by $d^o(X^0) = 0$, $d^o(X^{\bar{e}}) = 1$ and $d^o(X^{p+q}) = d^o(X^p) + d^o(X^q)$. To keep consistent with the usual terms, an *elementary test*, or test for short, is a monomial X^p with $d^o p = 1$ and a *linear form* is a finite sum $\sum_i e_i * X^{p_i}$ where $d^o p_i \leqslant 1$.

As usual, a polynomial is defined as a sum of monomials. The set of polynomials over $\mathbb{E}$ is defined as $\mathbb{E}[X]$ and ranged over by letters $P, Q, \ldots$ Given $P, Q \in \mathbb{E}[X]$, $P+Q$ and $P*Q$ are defined the usual way, thus providing $\mathbb{E}[X]$ with a commutative unitary ring structure.

Given a set S of polynomials, the ideal generated by S, written $\langle S \rangle$ is the set of polynomials $\sum T_i * P_i$, defined as finite sums where each $P_i \in S$.

2.2 Zeroes of Polynomials

If $ab = 0$ implies $a = 0$ or $b = 0$, the ring is said to be an integral domain; this is equivalent to $\langle 0 \rangle$ being a prime ideal. Rephrasing $a = 0$ as $a \in \langle 0 \rangle$, we extend the classical definition to $P(e) \in \pi$, where π is some proper prime ideal form $\mathbb{E}$. In the following lines, π will be assumed implicitly with the required conditions.

For $I \in \mathbb{E}[X]$ ideal, define $V(I) \equiv \{e \in \mathbb{E} | \forall P \in I.P(e) \in \pi\}$. To keep consistent with the usage in Algebraic Geometry, $V(I)$ is said to be an algebraic set. Conversely, given $S \subseteq \mathbb{E}$, define $I(S) \equiv \{T \in \mathbb{E}[X] \mid \forall e \in S.T(e) \in \pi\}$. Since π is an ideal, whichever S, the set $I(S)$ is an ideal of $\mathbb{E}[X]$. By construction, $\pi \subseteq I(S)$ is always true. Therefore, we shall always assume any polynomial ideal contains π. The following proposition has been sketched in the introduction. In the following, a ideal generated by (elementary) test is called a *tests ideal*. In this paper, we concentrate on these test ideals which appear to be sufficient; therefore, more general polynomials are not considered anymore in the sequel.

Proposition 3 (Zariski topology) *The collection of $V(I)$, where I ranges over all the test ideals from $\mathbb{E}[X]$ are the closed sets of a Zariski topology on $\mathbb{E}$.*

Note 1 By construction, any polynomial is continuous wrt his topology. So is $(\bullet e)$ for all $e \in \mathbb{E}$ as a particular case. This result compares with Ghilezan [12], where a topology on Λ is built (for each context Δ) with basis $(\mathcal{V}_{\Delta,\sigma})_{\sigma \in \mathrm{Type}}$ and $\mathcal{V}_{\Delta,\sigma} \equiv \{e \mid \Delta \vdash e : \sigma\}$. In both cases, λ-abstraction does not seem to be continuous. And as far as the application is concerned, the sets $\mathcal{V}_{\Delta,\sigma}$ could be read as being closed as well, with the same continuity result.

3 Generic interpretation

Let $R \subseteq \mathbb{E}$ sub-ring closed by reductions, and π prime and proper ideal of R, with respect to the induced ring structure. Define the following conditions

(stable R) For all $e \in \mathbb{E}$ such that $\mathrm{FV}(e) \subseteq \{x_1, \ldots, x_n\}$, $e(R^n) \subseteq R$.

(hereditary π) whenever $(e\bar{u}) \in \pi$ then $e \in \pi$.

(sat $\pi 0$) For all $\bar{u} \in \bar{R}$, $(0\bar{u}) \in \pi$.

(sat $\pi-$) Whenever $(a\bar{u}) \in \pi$ then $(-a\bar{u}) \in \pi$.

(sat $\pi+$) Whenever $(a\bar{u}) \in \pi$ and $(b\bar{u}) \in \pi$ then $(a + b\bar{u}) \in \pi$.

(sat $\pi*$) Whenever $(a\bar{u}) \in \pi$ and $(b\bar{u}) \in R$ then $(a * b\bar{u}) \in \pi$. (and the symmetric case.)

(sat $\pi\beta$) Whenever $(e[a/x]\bar{u}) \in \pi$ with $a \in R$ then $(\lambda x.ea\bar{u}) \in \pi$.

The saturation properties $(\mathrm{sat}\,\pi \ldots)$ come with no surprise for the reader. The stability condition $(\mathrm{stable}\,R)$ on the ring R, is required for the typing rule (ω). (hereditary π) ensures the non emptyness pf the following interpretation.

We first provide an interpretation of types along the lines of:

i. $[\omega] = R$ and $[t] = \pi$ for each type variable t;

ii. $[\sigma \wedge \tau] = [\sigma] \cap [\tau]$;

iii. $[\sigma \to \tau] = \{e \in \pi | \forall a \in [\sigma].(ea) \in [\tau]\} = [\sigma] \to [\tau]$.

The interpretation is extended to contexts by $[\varepsilon] = \{\mathrm{id}\}$ and $[\Delta, x : \sigma] = \{s[a/x] | s \in [\Delta], a \in [\sigma]\}$. Elements $s \in [\Delta]$ are substitutions, but we allow ourselves to use the applicative notation $e(s) = e(a_1, \ldots, a_n)$ as well.

Proposition 4 *For all σ, $[\sigma]$ is a R-ideal, such that $[\sigma] \subseteq \pi$ if $\sigma \neq \omega$. For all σ, τ, if $\sigma \leqslant \tau$ then $[\sigma] \subseteq [\tau]$.*

Given Δ, e and σ such that $\mathrm{FV}(e) \subseteq \mathrm{FV}(\Delta)$, define $\Delta \vDash e : \sigma$ if $e([\Delta]) \subseteq [\sigma]$. The interpretation of judgement $\Delta \vdash e : \sigma$ is thus well defined by $[\Delta \vdash e : \sigma] \equiv \Delta \vDash e : \sigma$.

Proposition 5 (Generic interpretation) *If $\Delta \vdash e : \sigma$ then $\Delta \vDash e : \sigma$.*

Proof Under the above assumption on the pair (R, π), this proof proceeds as usual. □

Note 2 Assume we want to use this framework to prove confluence. The condition $(\mathrm{stable}\, R)$ cannot be fulfilled unless we take $R = \mathbb{E}$. Then, the requirement that π is an ideal prevents us from considering type interpretations as subsets of terms which have the confluence property. Therefore, due to the presence of the multiplication $*$, we cannot apply the method suggested in [11], which justifies the direct proof in section 1.

In [14], Paolini and Ronchi Della Rocca propose a formal presentation of different λ-calculi, based on a notion of *input values*. Such a set is closed by substitutions involving values, closed by reduction and contains variables. The latter condition is not part of our conditions on R in general, but is satisfied in the next section.

4 Strong normalisation

Owing to the previous analysis, the strong normalisation issue for $\mathbb{E}$ can be dealt along the same lines as for pure λ-calculus. Let us consider the differences first.

Due to the presence of associative-commutative rules in $\mathbb{E}$, we cannot expect a straightforward notion of strong normalisation. Observe that given any term e, the set $\{e' | e \to^*_{\mathrm{AC}} e'\}$ is finite. Let us define $e \rhd e'$ iff $e \to^*_{\mathrm{AC}} e'$, and $e \Rightarrow e'$ if $e \to e'$ applying any reduction rule, but no associative or commutative one. Clearly, any reduction path $e \to^* e'$ can be described as an alternation of $\rhd$ and $\Rightarrow$ steps, in a unique way. Let $\ell(e \to^* e')$ stands for the number of $\Rightarrow$ steps in the reduction, and call this integer its *length*. Then $e \in \mathbb{E}$ is *strongly normalising* if there exists an upper bound $\|e\| \in \mathbb{N}$ to the lengths of reduction paths originated at e. This definition extends the classical one since $e \rhd e$ is the only possibility for pure λ-terms.

We restrict the typing rules by dropping $(T\omega)$. As a matter of consequence, type constant ω never occurs in any judgement $\Delta \vdash e : \sigma$ in the resulting system. Hence, condition $(\mathrm{stable}\, R)$ can be dropped as well. Notice moreover that $\Delta \vdash 1 : \sigma$ iff $\sigma = \omega$. Therefore our type assignment system does not accept the constant 1 any more, although it is strongly normalising according to the previous definition. Our position is that 1 cannot be accepted to have any particular property. This is consistent with the usual approaches where a special constant$\perp$ or $\mathbf{\Omega}$ is introduced. The difference is however that we do not use any ordered structure, as the reader may already have remarked.

Lemma 2 *The set R of strongly normalising terms plus* 1 *is a sub-ring of* $\mathbb{E}$.

Proof The subset of rules we chose for algebraic reductions prevents us from using a (more) simple polynomial interpretation. Our choice seemed more naturral wrt mathematical usage, and requires instead the usual recursive path ordering on which it is then possible to reason inductively. □

Let us consider the R-ideal π_0 generated by the terms $(x\bar{u})$ with $\bar{u} \in \bar{R}$. This ideal is well defined and proper, could be prime as well, but is not guaranteed to satisfy the former saturation properties. We take for π some maximal ideal which includes π_0 and avoid the multiplicative set $\{1\}$. This ideal exists as a general result of Abstract Algebra. Moreover,

Lemma 3 *The ideal π is a prime and proper R-ideal which satisfies the saturation conditions.*

Proof π is maximal hence prime. $1 \notin \pi$ by construction: π is proper. Let us consider the condition $(\mathrm{sat}\,\pi\beta)$. Take $a \in R$ such that $(e[a/x]\bar{u}) \in \pi$. If $(\lambda x.ea\bar{u}) \notin \pi$, since π is maximal, there exists $p \in \pi$ and $b \in R$ such that $1 = (\lambda x.ea\bar{u}) * b + p$. But then $1 = (e[a/x]\bar{u}) * b + p \in \pi$ which is absurd. Other cases are similar. □

Proposition 6 *If $\Delta \vdash e : \sigma$ within the restricted type assignment system, then $e \in R$, hence e is strongly normalising.*

Proof First $[\sigma] \subseteq \pi$ is always true in that case. By proposition 5, we get $e([\Delta]) \subseteq [\sigma] \subseteq \pi \subseteq R$. We can check also that $\pi_0 \subseteq [\sigma]$ which means that every $[\sigma]$ has variables. Thus $e \in R$ as well. □

5 The Full Abstraction Problem

In this section, we are concerned with programs, which means that we interpret our calculus in the set $\mathbb{E}^0$of closed terms. Clearly, $\mathbb{E}^0$ is a ring; so we take $R \equiv \mathbb{E}^0$. According to some evaluation mechanism, we expect programs to be evaluated to values. Following the classical approach, we may define the set $\pi_0 \subseteq R$ of *values* according to the grammar:

$$v \quad ::= \quad \lambda x.e | e * v | v * e | - v | v + v | 0$$

but it should be clear that π_0 is defined equivalently as the ideal generated by the set of (closed) λ-abstractions.

We now define the evaluation of programs using notation $e \Downarrow v$ is such a way that saturation conditions are actually part of the definition of π:

$$\begin{array}{ll}
(\text{Value}) & \dfrac{v \in \pi_0}{v \Downarrow v} \\
(\text{App} - \lambda) & \dfrac{(e[a/x]\breve{u}) \Downarrow v}{(\lambda x.ea\bar{u}) \Downarrow v} \\
(\text{App} - 0) & \dfrac{(0\bar{u}) \in R}{(0\bar{u}) \Downarrow 0} \\
(\text{App} - +) & \dfrac{(a\bar{u}) + (b\bar{u}) \Downarrow v}{(a + b\bar{u}) \Downarrow v} \\
(\text{App} - -) & \dfrac{-(a\bar{u}) \Downarrow v}{(-a\bar{u}) \Downarrow v} \\
(\text{App} - *) & \dfrac{(a\bar{u}) \Downarrow v(b\bar{u}) \in R}{(a * b\bar{u}) \Downarrow v} \text{and the symmetric case}
\end{array}$$

We say that the program e ***has a value*** or ***converges***, denoted $e \Downarrow$, if there exists $v \in \pi_0$ such that $e \Downarrow v$ according to the previous rules. Otherwise, the program e ***diverges***, $e \Uparrow$ for short. Let us notice that 1 is both closed and divergent according to these definitions.

Lemma 4 *Let $\pi \equiv \{e \in R | \exists v \in \pi_0 . e \Downarrow v\}$. Then π is a prime and proper ideal of R.*

Proposition 7 (Convergence)

$$\forall e \in R \qquad \vdash e : \omega \to \omega \textit{ iff } e \in \pi \text{ iff } e \Downarrow \tag{1}$$

Proof Proposition 5 provides the necessary condition. To prove the condition is sufficient, we proceed by induction on the evaluation rules; as for the base case, by induction structural on values. □

5.1 Canonical elements

The generic model shows the $[\sigma] = V(I)$ for some ideal I which is generated by elementary tests. A priori, I is not finitely generated, due to the interpretation of the arrow type. Following [1, 6] we look for a family $(e_\sigma, \nabla_\sigma)_{\sigma \in \mathsf{Type}}$, where $e_\sigma \in R$ and $\nabla_\sigma \in R[X]$, such that

I. $\vdash e_\sigma : \tau$ iff $\sigma \leqslant \tau$;

II. $\forall e \in R . \nabla_\sigma(e) \Downarrow$ iff $\vdash e : \sigma$.

These two conditions express the fact that both sets of rules which occur on the syntactical side ($\sigma \leqslant \tau$ and $e : \sigma$ respectively) are expressible in the calculus itself. As far as convergence is concerned, we *need* also to represent the predicate $e \Downarrow$.

Let us have a closer look at property II. It asserts the existence of $\nabla_{\omega \to \omega}(X)$ such that $\nabla_{\omega \to \omega}(e) \Downarrow$ iff $\vdash e : \omega \to \omega$. We have already $\pi = V(\langle X \rangle)$, and the test X should be sufficient for this purpose. Actually, we need a bit more, since each ∇_σ carries also the property: if $\nabla_\sigma(e) \Downarrow$ then $\nabla_\sigma(e) \Rightarrow \boldsymbol{I}$, where $\boldsymbol{I}$ is the identity combinator.

Assume the existence of $\nabla(X) \in R[X]$ such that

$$\nabla(\bullet) \text{is a ring homomorphism} : \mathbb{E} \to R \,\text{with kernel}\, \pi \tag{2}$$

$$\forall e \in R \qquad \nabla(e) \Downarrow \text{iff} \vdash e : \pi \tag{3}$$

Then we define the elements $(e_\sigma, \nabla_\sigma)$ by induction on σ as

- $e_\omega \equiv 1$ and $\nabla_\omega(X) \equiv \boldsymbol{I}$;
- $e_{\sigma\to\tau} \equiv \lambda x.(\nabla_\sigma(x)e_\tau)$ and $\nabla_{\sigma\to\tau}(X) \equiv \nabla(X) + \nabla_\tau(Xe_\sigma)$;
- $e_{\sigma\wedge\tau} \equiv e_\sigma * e_\tau$ and $\nabla_{\sigma\wedge\tau}(X) \equiv \nabla_\sigma(X) + \nabla_\tau(X) - \boldsymbol{I}$.

Proposition 8 *The family $(e_\sigma, \nabla_\sigma)_{\sigma\in \mathsf{Type}}$ satisfies conditions I and II.*

Proof Follows [10], proving the more precise condition (II') stating $\vdash \nabla_\sigma(e) : \tau \to \gamma \to \delta$ iff $\sigma \leqslant \tau$ and $\gamma \leqslant \delta$. □

Proposition 9 (Completeness) *For all Δ, σ and $e \in \mathbb{E}$ such that $\mathrm{FV}(e) \subseteq \mathrm{FV}(\Delta)$, $\Delta \vdash e : \sigma$ iff $e([\Delta]) \subseteq [\sigma]$.*
Proof *$\Rightarrow$ results for the tuple (R, π) satisfying the conditions of the generic model. The converse part is proved, once again, as in [10].* □

Towards full abstraction, we restrict ourselves to a precise notion of observation context. Given $e \in \mathbb{E}$ with $\mathrm{FV}(e) \equiv \{x_1, \ldots, x_n\}$, a valid test for e is any context $C[] \equiv P([](s))$ where $P(X) \in R[X]$ and s substitution such that $s(x_i) \in R$ for each $i = 1, \ldots, n$. The chosen definition for tests may seem more restrictive, but it is already known that these contexts are actually sufficient (see for example the detailed proofs of [4]). Let us define on $\mathbb{E}$ the pre-orders

Typability $a \sqsubseteq^T b$ iff for all Δ, σ $\Delta \vdash a : \sigma$ implies $\Delta \vdash b : \sigma$;

Observation $a \sqsubseteq^O b$ iff $C[a] \Downarrow$ implies $C[b] \Downarrow$ for any valid test $C[]$.

Let $\Gamma \vdash s : \Delta$ for $\Gamma \vdash s(x) : \sigma$ for all $(x : \sigma) \in \Delta$. Then

Lemma 5 (Paste) *If $\Gamma \vdash e(s) : \sigma$, then $\exists\Delta.\Gamma \vdash s : \Delta$ and $\Delta \vdash e : \sigma$.*

Proof By induction on the derivation tree $\Gamma \vdash e(s) : \sigma$. □

Theorem 1 (Full abstraction) *$\sqsubseteq^T = \sqsubseteq^O$.*

Proof If $\Delta \vdash a : \sigma$ and $\Delta \nvdash b : \sigma$, we show the test $C[] = \nabla_\sigma([]s_\Delta)$ where s_Δ is defined by $s_{\Gamma,x:\tau} \equiv s_\Gamma, e_\tau$, separates a and b: $C[a] \Downarrow$ and $C[b] \Uparrow$. Conversely, the proof is by induction on $P ::= e|(X\bar{u})|P * P|P + P$ using completeness and the fact $[\omega \to \omega] = \pi$ is a prime ideal and the previous (Paste) lemma. □

Note 3 In [4, 10], call-by-value is introduced in the syntax. Although for different purposes, this is required in both cases in order to establish full abstraction results. The delayed introduction of the element $\nabla \in R[X]$, and thus in $\mathbb{E}$ as a matter of consequence, is not intended to argue that such a feature is useless. We rather expect to get a closer look at the precise mechanism which is required toward this result. In fact, $\nabla(a) + b$ expresses the requirement that computations of a converges, while it diverges otherwise, whichever the computation of b. One could have introduced another operator as well: say $\nabla'(e)$ reducing to $\boldsymbol{I}$ in spite of 0. Then $(\nabla'(a)b)$ would express the fact the computations are made in *sequence*. The latter has the same operational behaviour as the construct $\mathrm{seq}(a; b)$ presented in [5]. As far as no side effect is added to the calculus, the effect of the former term is the same, *without* constraining any order on the way computations are done.

Since the the full abstraction results requires the introduction of the ∇ operator, for the proofs to be valid, one needs first extending the syntax with the new term constructor $\nabla(e)$, where $e \in \mathbb{E}$. This introduction requires also additional reduction rules. The whole system is presented and studied in the full version of this abstract.

Conclusion

The generic model presented in section 3 provides a uniform framework for the study of term properties. The attentive reader may already have observed that the full apparatus is not absolutely required for the current purpose, since type interpretation does not involve more than elementary monomials $(X\bar{u})$, which make them appear as simple as hyper-plan intersections. Actually, our motivations for introducing right now the full structure are manifold.

As shown in the introduction, the parallel operator appears quite naturally, introduced by our topological analysis, and suggests a different approach to concurrency, where terms are pieces of computation which are combined along the universal rules of algebra: terms can be added or subtracted (reversed) as in any group structure. Product of terms leads us, along the same lines as for Algebraic Geometry, to localisation of computations by making terms of less interest (for instance divergent programs) invertible. The classical technique of localisation of a ring has a intuitive meaning in the present context: informally, this can be viewed as expressing the idea that it is possible to *see* locally a program as convergent, even terminating, while it is a fragment of a bigger program which can be divergent as well: running the Unix program `date` at the shell prompt is a suggestive, although not necessarily accurate example of such a situation.

Our calculus is based on the idea that computation should be allowed in each type σ, hence the rules which ensure σ has an additive group structure. This syntactic presentation has strong consequences, and allow a quite simple interpretation for types. In the meantime, this approach leads to a very coarse analysis with respect to terms. For example, the least closed set which contains a term e exists in the model: it is the algebraic set built from all the polynomials cancelled by e and is well known as the closure of e. Of course we cannot restrict ourselves to elementary monomials anymore. Type assignment systems do not provide a *type* for this *generic point*, while the present presentation offer a richer notion of type, as far as term properties are concerned.

The present extended calculus is actually the result of our reading of [4], toward an attempt to clarify the many steps of these proofs, their relationships, and also to come closer to the point where the requirements for *call-by-value* abstraction for one side, and *parallel operator* for the other were required. Although we agree call-by-value comes quite naturally in programming languages (as well as in mathematics), our various readings do not prove it is strictly necessary by itself as far as full abstraction is concerned: in [10, 4] this construction is only required for tests, not for characteristic terms strictly speaking. So we were convinced it should be worth narrowing the very place where something more in order to get completeness. The present approach aimed at showing that this result is part of one-to-one correspondence between sets of terms (algebraic sets) and polynomial ideals. Our presentation expresses, with no formal proof, that we need the existence of some homomorphism ∇ which is not representable in the initially defined ring $\mathbb{E}[X]$. Comparison with algebraic geometry may provide interesting guidelines or more, toward deeper models of computation.

Another observation comes to mind. Returning to section 5, the pair formed by π (interpretation of $\omega \rightarrow \omega$) and the multiplicative set of divergent terms play with respect to R a rôle similar to the pair $[D \rightarrow D], \perp$ wrt a Scott domain solution D of the equation $D = [D \rightarrow D]_\perp$ in Abramsky's interpretation of the lazy λ-calculus. This observation strongly emphasises an alternate approach outside ordered structures. We expect to put this observation at work through the use of the (ring of) approximants; still a convenient notion net of terms require some work in order to express accordingly the proper idea of approximants.

As a whole, we feel this framework is worth being studied more deeply and we intend to address the above points more precisely.

Acknowledgements

The author wants to thank the referee from TLCA'05 for their interest in the matter, noticing misprints and providing both suggestions for improvement and additional references. Many thanks due also to Yves Bertot, Loïc Pottier and Laurence Rideau for feedback and encouragements.

References

[1] S. Abramsky. Domain theory in logical form. In *Second IEEE Symposium on Logic in Computer Science*, 1987.

[2] S. Abramsky. The lazy lambda-calculus. In D. Turner, editor, *Research Topics in Functional Programming*, pages 65–116. Addison-Wesley, 1989.

[3] Philippe Audebaud. Explicit substitutions for the lambda-mu calculus. Technical report, Laboratoire de l'Informatique du Parallélisme, 1994.

[4] Gérard Boudol. Lambda-calculi for (strict) parallel functions. *Information and Computation*, 108(1), 1994.

[5] R. L. Constable and S. F. Smith. Partial objects in constructive type theory. In *Second IEEE Symposium on Logic in Computer Science*, 1987.

[6] M. Coppo. Completeness of type assignment in continuous lambda models. *tcs*, 29:309–324, 1984.

[7] Mario Coppo, Mariangiola Dezani-Ciancaglini, and Furio Honsell. Applicative information systems. In *CAAP'83*, volume 159 of *lncs*, 1983.

[8] Mario Coppo, Mariangiola Dezani-Ciancaglini, Furio Honsell, and Giuseppe Longo. Extended type structures and filter lambda models. In G. Lolli, G. Longo, and A. Marcja, editors, *Logic Colloquium '82*, 1983.

[9] Vincent Danos and Jean-Louis Krivine. Disjunctive tautologies as synchronisation schemes. In *Proceedings of the 14th Annual Conference of the EACSL on Computer Science Logic*, pages 292–301. Springer-Verlag, 2000.

[10] M. Dezani-Ciancaglini, U. de' Liguoro, and A. Piperno. "A filter model for concurrent lambda-calculus". *Siam Journal on Computing*, 27(5):1376–1419, 1998.

[11] S. Ghilezan and S. Likavec. Extensions of the reducibility method. In *4th Panhellenic Logic Symposium (PLS4)*, 2004.

[12] Silvia Ghilezan. Full intersection types and topologies in lambda calculus. *Journal of Computer and System Sciences*, (62):1–14, 2001.

[13] Randall Holmes. Notes on positive set theory. unpublished material, 2001.

[14] Luca Paolini and Simona Ronchi Della Rocca. Parametric parameter passing λ-calculus. *Inf. Comput.*, 189(1):87–106, 2004.

[15] Michel Parigot. Strong normalization for second order classical natural deduction. In *Eigth IEEE Symposium on Logic in Computer Science*, 1993.

[16] Mike Smyth. Powerdomains and predicate transformers: a topological view. In J. Diaz, editor, *Automata, Languages and Programming*, volume 154 of *lncs*. sv, 1983.

[17] M.H. Stone. Topological representation of distributive lattices and brouwerian logics. *Casopis Pest. Mat. Fys.*, 67:1–25, 1937.

[18] A. Tarski. Der aussagenkalkül und die topologie. *Fund. Math.*, 31:103–134, ??

[19] Steven Vickers. *Topology via logic*. Cambridge University Press, 1989.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399